

University of Groningen

MultiLevel simultaneous component analysis

Ceulemans, Eva; Wilderjans, Tom F.; Kiers, Henk A.L.; Timmerman, Marieke E.

Published in:
Behavior Research Methods

DOI:
[10.3758/s13428-015-0626-8](https://doi.org/10.3758/s13428-015-0626-8)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Early version, also known as pre-print

Publication date:
2016

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Ceulemans, E., Wilderjans, T. F., Kiers, H. A. L., & Timmerman, M. E. (2016). MultiLevel simultaneous component analysis: A computational shortcut and software package. *Behavior Research Methods*, 48(3), 1008-1020. <https://doi.org/10.3758/s13428-015-0626-8>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

MultiLevel simultaneous component analysis: A computational shortcut and software package

Eva Ceulemans^{1,4} · Tom F. Wilderjans^{1,2} · Henk A. L. Kiers³ · Marieke E. Timmerman³

© Psychonomic Society, Inc. 2015

Abstract MultiLevel Simultaneous Component Analysis (MLSCA) is a data-analytical technique for multivariate two-level data. MLSCA sheds light on the associations between the variables at both levels by specifying separate submodels for each level. Each submodel consists of a component model. Although MLSCA has already been successfully applied in diverse areas within and outside the behavioral sciences, its use is hampered by two issues. First, as MLSCA solutions are fitted by means of iterative algorithms, analyzing large data sets (i.e., data sets with many level one units) may take a lot of computation time. Second, easily accessible software for estimating MLSCA models is lacking so far. In this paper, we address both issues. Specifically, we discuss a computational shortcut for MLSCA fitting. Moreover, we present the MLSCA package, which was built in MATLAB, but is also available in a version that can be used on any Windows computer, without having MATLAB installed.

Keywords Multilevel data · Component analysis · Software

Introduction

Nowadays, more and more studies result in multivariate two-level data. In sensometric research, for instance, such data are obtained when multiple panelists each rate different sets of food samples on a number of descriptor variables (e.g., Bro, Qannari, Kiers, Naes, & Frøst, 2008). Take as a second example an emotion psychologist who studies how the experience of multiple emotions evolves across time through experience sampling, and does this for a number of subjects (e.g., Erbas, Ceulemans, Pe, Koval, & Kuppens, 2014). Such data have a two-level structure, in that the observation units (i.e., food samples, measurement occasions) can be considered level one units that are nested within level two units (i.e., panelists, subjects). Moreover, the data are multivariate in that all level one units are measured on multiple variables. In the remainder, following De Roover, Ceulemans and Timmerman (2012), we will denote the level two units by “data blocks” and the level one units by “observations.”

Many research questions regarding two-level multivariate data pertain to the associations between the variables, which need not be the same across all data levels. In case the variables fall apart in predictor (i.e., independent) and criterion (i.e., dependent) variables, multilevel regression models (Snijders & Bosker, 1999) may be adopted to properly model these different associations. In case no such distinction is made and the number of variables is somewhat larger (say, larger than five), researchers often turn to multilevel factor analysis. The key principle behind this method is that the observed variables are considered to be indicators of latent variables, which account for the associations between the variables. Two types of multilevel factor analysis techniques exist. The first type is multilevel confirmatory factor analysis (multilevel CFA), which aims at testing whether specific hypotheses regarding the factor structure are consistent with the data (e.g., Mehta & Neale, 2005). The second type is

✉ Eva Ceulemans
eva.ceulemans@ppw.kuleuven.be

¹ University of Leuven – KU Leuven, Leuven, Belgium

² Leiden University, Leiden, The Netherlands

³ Heymans Institute for Psychology, DPMG, University of Groningen, Groningen, The Netherlands

⁴ Methodology of Educational Sciences Research Group, Tiensestraat 102, B-3000 Leuven, Belgium

exploratory in nature (multilevel EFA) and is used to find the latent factors that best fit the data at hand (e.g., Goldstein & Browne, 2005).

In multilevel EFA, the variance in the data is split in a between part, containing the means of each data block, and a within part, consisting of the deviations from the block means, and separate exploratory factor models are fitted to each part. Hence, between-block differences in variable means and covariance structure can be studied separately. Multilevel EFA, however, has the disadvantage that it requires a large number of data blocks.

An interesting alternative to multilevel EFA was provided by Timmerman (2006), who proposed a class of multilevel simultaneous component models (MLSCA). Like a multilevel EFA model, a MLSCA model sheds light on the associations between the variables at the different levels, by specifying separate submodels for the level two units and the level one units. Each submodel consists of a component model. MLSCA has already been successfully applied to study individual differences in development (Timmerman, Ceulemans, Lichtwarck-Aschoff, & Vansteelandt, 2009) and attachment (Bosmans, Van de Walle, Goossens, & Ceulemans, 2014), cross-cultural differences (Kuppens, Ceulemans, Timmerman, Diener, & Kim-Prieto, 2006), and differences between experimental conditions in a social dilemma game (Stouten, Ceulemans, Timmerman, & Van Hiel, 2011) and genomics (Jansen, Hoefsloot, van der Greef, Timmerman, & Smilde, 2005).

However, the use of MLSCA is hampered by two issues. First, as MLSCA solutions are obtained through iterative algorithms, analyzing large data sets (i.e., data sets with many observations) may take a lot of computation time. Second, whereas for multilevel regression, multilevel CFA, and multilevel EFA easily accessible software exists (e.g., MPLUS, SAS, and R-packages), up to now, such software for estimating MLSCA models has been lacking.

In this paper, we address both issues. Specifically, we discuss a computational shortcut for MLSCA fitting that considerably decreases computation time in cases where the number of level one units is much larger than the number of variables. Moreover, we present the MLSCA package, which was built in MATLAB, but is also available in a version that can be used on any Windows computer, without having MATLAB installed. This package may be freely downloaded from <http://ppw.kuleuven.be/okp/software/MLSCA>. The MLSCA package allows the user to interact with the program through a graphic user interface, in which different analysis options can be specified (e.g., different number of between- and within-components and different MLSCA variants). Through built-in model selection heuristics, the package assists the user in selecting a model that fits the data well and is parsimonious at the same time. The analysis results are automatically saved in formats that can be easily loaded in popular software packages (e.g., SAS and SPSS) to further process them (e.g.,

plotting the component scores/loadings, correlating the obtained estimates with other variables).

The remainder of this paper is organized into four main sections. In the second section, MLSCA theory is recapitulated. In the third section, the different steps of an MLSCA analysis are discussed and demonstrated making use of sensory profiling data. In this section, we also present the computational shortcut for large data sets. The fourth section introduces the MLSCA software package. Finally, the fifth section contains some concluding remarks.

MultiLevel simultaneous component analysis (MLSCA)

Data structure and preprocessing

MLSCA models a data matrix \mathbf{X} , that consists of I columnwise concatenated data blocks \mathbf{X}_i ($i = 1 \dots I$), consisting of K_i observations by J variables. \mathbf{X} can be split uniquely into three parts – an offset term, a between-part, and a within-part – as:

$$\mathbf{X} = \mathbf{X}^{\text{offset}} + \mathbf{X}^{\text{between}} + \mathbf{X}^{\text{within}}. \quad (1)$$

The entries of these three matrices are obtained by decomposing the score x_{ijk_i} of observation k_i within data block i on variable j :

$$\begin{aligned} x_{ijk_i} &= x_{ijk_i}^{\text{offset}} + x_{ijk_i}^{\text{between}} + x_{ijk_i}^{\text{within}} \\ &= x_{.j} + (x_{ij.} - x_{.j}) + (x_{ijk_i} - x_{ij.}), \end{aligned} \quad (2)$$

with $x_{.j}$ denoting the mean score on variable j computed across all data blocks, and $x_{ij.}$ indicating the mean score on variable j computed within data block i .

Since component analysis identifies components that capture as much of the variance in the data as possible, variance differences among the variables may hugely affect the obtained results. Therefore, researchers should decide whether or not such variance differences are meaningful. For instance, when analyzing physiological measures such as heart rate, respiratory volume, and blood pressure, variance differences are at least partly arbitrary because the variables are measured on a different scale (see e.g., De Roover, Timmerman, Van Diest, Onghena, & Ceulemans, 2014). Thus, it makes sense to give each variable the same weight in the analysis by scaling each variable to a variance of one across all data blocks. However, as a counterexample, when studying emotions in daily life, the variances of negative emotions are often much smaller than those of positive emotions, even though they are rated using the same scale. Such differences are meaningful because negative emotions are experienced less often than positive ones. Discarding these variance differences may yield misleading

results (see e.g., Brose, De Roover, Ceulemans, & Kuppens, 2015); therefore, one should not scale the variables to equal variance in this case.

Model

The full MLSCA model for the observed data is obtained by summing the offset term and the between- and within-submodels (see Timmerman, 2006), which we discuss below.

Between-submodel

The between-submodel captures the differences between data blocks in variable means. It boils down to an ordinary principal component analysis (PCA) of $\mathbf{X}^{between}$. Formally, each of the I between-parts $\mathbf{X}_i^{between}$ is approximated by $\hat{\mathbf{X}}_i^{between}$ which is decomposed as follows:

$$\hat{\mathbf{X}}_i^{between} = \mathbf{1}_{K_i} \mathbf{f}_i^b \mathbf{B}^b{}^T, \quad (3)$$

with $\mathbf{1}_{K_i}$ being a $K_i \times 1$ vector of ones, \mathbf{f}_i^b ($1 \times Q^b$) containing the between-component scores of data block i on the Q^b between-components, and \mathbf{B}^b ($J \times Q^b$) being the between-component loading matrix. Per component, the overall mean (i.e., across all blocks) of the between-component scores equals zero, and its overall variance amounts to one.

Two remarks regarding the between-loadings are in order. First, these loadings may be orthogonally or obliquely rotated, provided that the between-component scores are counterrotated. This rotational freedom is often exploited to facilitate the interpretation of the obtained components, by rotating the loadings towards simple structure. Second, a raw between-loading amounts to the covariance among the between-component and the between-part of the observed variable. To ease interpretation, one can compute normalized loadings, which can be read as correlations rather than covariances, by dividing each raw loading by the standard deviation of the corresponding column of $\mathbf{X}^{between}$.

Within-submodel

The within-submodel accounts for the covariance structure of the variables within the data blocks. More specifically, each \mathbf{X}_i^{within} is approximated by $\hat{\mathbf{X}}_i^{within}$ which is decomposed as:

$$\hat{\mathbf{X}}_i^{within} = \mathbf{F}_i^w \mathbf{B}_i^w{}^T, \quad (4)$$

with Q^w denoting the number of within-components for data block i , \mathbf{F}_i^w ($K_i \times Q^w$) being its within-component score matrix, and \mathbf{B}_i^w ($J \times Q^w$) its within-component loading matrix. Per block and per component, the mean of the within-component scores equals zero.

The most general variant, called MLCA, boils down to a separate PCA per \mathbf{X}_i^{within} , with the variance of the within-component scores being restricted to one for each of the block-specific components. Thus, each data block has its own component loading matrix \mathbf{B}_i^w , implying that the within-component scores of the different data blocks cannot be compared. Each loading matrix \mathbf{B}_i^w can be separately rotated. MLCA should be used when equal within-loadings for the different data blocks are implausible. For instance, when assessing the structure of values in different cultures, it is reasonable that this structure differs depending on, amongst others, the industrialization rate of the cultures (see e.g., De Roover, Timmerman, Mesquita, & Ceulemans, 2013).

However, for many data sets it makes sense to expect that the within-loadings of the data blocks are equal or at least very similar. Then, the structure is (about) the same, but the variances and/or correlations of the component scores may still differ across data blocks. As an example, Erbas et al. (2013) studied the structure of 20 emotions in typically developing adolescents and in adolescents with autism spectrum disorder (ASD). They found that the emotion loadings are very similar in both groups, but that the emotion components are more strongly correlated in the autism group. As another example, Brose et al. (2015) summarizes multiple studies on daily affect in different age groups by stating that the level of negative affect varies less across days in older adults. To model such data, based on the simultaneous component analysis (SCA) framework of Timmerman and Kiers (2003), Timmerman (2006) distinguished four different model variants (see Table 1), which imply different sets of restrictions on the \mathbf{F}_i^w matrices. Note that all variants constrain \mathbf{B}_i^w to be the same for all data blocks.

The four variants can be ordered in terms of restrictiveness. MLSCA-P is the least restrictive variant in that no further constraints are imposed on the variances and correlations of the within-component scores. When these correlations are constrained to have the same absolute values in all data blocks, but the variances are left untouched, MLSCA-PF2 is obtained. Importantly, this constraint on the correlations implies that the signs of these correlations may differ across the blocks (see Helwig, 2013), which has not always been acknowledged in previous papers on MLSCA. If such sign differences are unreasonable and occur, a constrained MLSCA-PF2 version can be considered, which restricts the correlations to be exactly the same. This model can be called MLSCA-PF2-NN, as it imposes non-negativity constraints on a further decomposition of the \mathbf{F}_i^w matrices (see Harshman, 1972; Kiers, ten Berge & Bro, 1999). MLSCA-IND goes one step further in this direction in that the correlations of the within-component scores are set to zero for each data block. Finally, in MLSCA-ECP, the most restrictive variant, both the variances and the correlations of the within-component scores are restricted to be the same for each data block. Note that in all four variants, the overall variances of the within-component scores equal one per component.

Table 1 Overview of the restrictions imposed on the variances (var) and correlations (corr) of the within-component scores of the separate data blocks by the different MultiLevel Simultaneous Component

Variant	Var	Corr	Complexity c_{Q^w}
MLSCA-P	None	None	$KQ^w + JQ^w - (Q^w)^2 - IQ^w$
MLSCA-PF2	None	Equal ^a	$KQ^w + JQ^w - (I+1)Q^w - (I-1)\frac{Q^w(Q^w-1)}{2}$
MLSCA-IND	None	Zero	$KQ^w + JQ^w - (I+1)Q^w - I\frac{Q^w(Q^w-1)}{2}$
MLSCA-ECP	Equal	Equal	$KQ^w + JQ^w - (I-1)Q^w - (I-1)\frac{Q^w(Q^w-1)}{2} - (Q^w)^2 - IQ^w$
MLCA	-	-	$KQ^w + IJQ^w - I(Q^w)^2 - IQ^w$

^a The unconstrained MLSCA-PF2 only imposes equality of the absolute values of the correlations

Two of these four variants, MLSCA-P and MLSCA-ECP, have rotational freedom. Specifically, the within-loadings can be rotated across data blocks, as long as these rotations are compensated for in the within-component scores. Note that a raw within-loading equals the covariance between the corresponding within-component and the within-part of the associated observed variable. Again, when the correlation, rather than the covariance, is of interest, the within-loadings can be normalized into correlations by dividing them by the standard deviation of the associated column of \mathbf{X}^{within} .

Steps in an MLSCA analysis

In this section, we discuss the three main steps of an MLSCA analysis: (1) fitting the different MLSCA variants, (2) model selection, that is, determining the optimal number of between- and within-components and the most adequate model variant for the within-part, and (3) interpreting the component matrices of the retained solution. These steps will be illustrated by analyzing a sensory profiling data set.

An important goal in sensory profiling research is to reveal whether panelists have systematically higher or lower ratings than other panelists; whether panelists, implicitly or explicitly, take different product features into account when judging food samples; whether panelists attach different weights to these product features; and whether these product features are differently associated across panelists. These four questions pertain to between-panelist differences in mean levels, within-component loadings, and the variances and correlations of the within-component scores, and can thus be answered using MLSCA.

To demonstrate this, we will analyze sensory profiling data concerning cream cheeses. Specifically, eight panelists were asked to rate three samples of ten different cream cheeses (i.e., 30 samples) with respect to 23 attributes, such as sweet,

Analysis (MLSCA) variants, and of the associated complexity values, with $K = \sum_{i=1}^I \min(K_i, \ln(K_i)J)$

grainy, and chalky (for a detailed description of the data set, see Bro et al., 2008). We consider the samples to be the level one units, which are nested within the panelists. In other words, the data blocks pertain to the sample by attribute data matrices of the different panelists. In accordance with previous analyses of these data (Bro et al., 2008; Ceulemans, Timmerman, & Kiers, 2011; De Roover, Timmerman, Van Mechelen, & Ceulemans, 2013; Wilderjans & Ceulemans, 2013), we opted to discard all variance differences between the attributes by scaling each of them to a variance of one across all data blocks (preprocessing option of the software). This implies that possible differences in variances between panelists are retained. The software package includes an excel-file that shows the raw data, the scaled version, and the splitting into offset term, between-part, and within-part.

Fitting the different MLSCA variants

Loss function

When fitting the different MLSCA variants for specific numbers of between-components and within-components, we look for estimates of \mathbf{f}_i^b , \mathbf{B}^b , \mathbf{F}_i^w , and \mathbf{B}_i^w that satisfy the imposed constraints and minimize the following loss function:

$$f = \sum_{i=1}^I \left\| \left(\mathbf{X}_i - \mathbf{X}_i^{offset} \right) - \left(1_{K_i} \mathbf{f}_i^b \mathbf{B}^b + \mathbf{F}_i^w \mathbf{B}_i^w \right) \right\|^2, \quad (5)$$

where $\mathbf{X}_i - \mathbf{X}_i^{offset}$ implies that we analyze the grand-mean centered data. Timmerman (2006) demonstrated that minimizing this loss function (Eq. 5) is equivalent to maximizing the percentage of variance in the data that the between- and within-components account for:

$$VAF\% = 1 - \frac{f}{\sum_{i=1}^I \left\| \mathbf{X}_i - \mathbf{X}_i^{offset} \right\|^2} \quad (6)$$

Furthermore, the VAF% may be computed for each submodel separately, because the offset, between-submodel, and within-submodel are mutually orthogonal. Specifically, the percentages of between- and within-variance accounted for amount to:

$$\begin{aligned} VAF_{\%}^{between} &= \frac{\sum_{i=1}^I \|1_{K_i} \mathbf{f}_i^b \mathbf{B}^b\|^2}{\sum_{i=1}^I \|\mathbf{X}_i^{between}\|^2} \\ VAF_{\%}^{within} &= \frac{\sum_{i=1}^I \|\mathbf{F}_i^w \mathbf{B}_i^w\|^2}{\sum_{i=1}^I \|\mathbf{X}_i^{within}\|^2}. \end{aligned} \quad (7)$$

Note that it is instructive to examine and compare the values of $\sum_{i=1}^I \|\mathbf{X}_i^{between}\|^2$ and $\sum_{i=1}^I \|\mathbf{X}_i^{within}\|^2$, as they indicate how much of the variance is situated at the between-level and how much at the within-level. For instance, these values equal 1,902.05 and 3,617.95 for the cheese data, implying that 34.46 % of the total variance is between-variance. Moreover, it is also worthwhile to inspect the amount of between- and within-variance for each variable separately.

Algorithm

Because the $\mathbf{X}_i^{between}$ and \mathbf{X}_i^{within} matrices are mutually orthogonal, the parameters of the between- and within-submodels can be estimated separately. In particular, \mathbf{f}_i^b and \mathbf{B}^b are obtained by conducting a singular value decomposition (SVD) on the vertically concatenated $\mathbf{X}_i^{between}$ matrices (for details, see Timmerman, 2006). The within-component scores and loadings of the MLCA and MLSCA-P models are estimated by an SVD of each separate \mathbf{X}_i^{within} matrix and of their vertical concatenation, respectively (Kiers & ten Berge, 1994a; Timmerman & Kiers, 2003). Because of the additional restrictions in the MLSCA-PF2, MLSCA-IND, and MLSCA-ECP models, an alternating least squares algorithm is required to estimate their associated within-component scores and loadings. These algorithms alternate between estimating \mathbf{B}^w and the separate \mathbf{F}_i^w matrices, starting from an initial configuration for \mathbf{B}^w (see Timmerman & Kiers, 2003). In case of MLSCA-PF2 analyses, it may happen that the obtained solution is degenerate in that the within-component scores are extremely strongly correlated in each data block. This degeneracy problem is well known in PARAFAC analysis where one possible solution is to impose orthogonality constraints (Harshman & De Sarbo, 1984). Following this line of reasoning, in case of a MLSCA-PF2 degeneracy, we recommend considering orthogonality restrictions and thus using MLSCA-IND. Moreover, to obtain MLSCA-PF2-NN solutions, non-negativity constraints can be imposed in a particular step of the alternating procedure (see, Kiers, ten Berge & Bro, 1999).

Given that the results of an ALS algorithm may depend on the starting configuration and thus may pertain to a local

optimum only, we recommend using a multi-start procedure. Specifically, we advise running the algorithm using a more or less rational start, consisting of the \mathbf{B}^w that is obtained through an MLSCA-P analysis, as well as using a number of randomly generated \mathbf{B}^w matrices, and retaining the best solution encountered across the different runs. When analyzing the cheese data, we used 100 random starts.

Computational shortcut

For large data sets, computation time can be considerable due to, amongst others, the SVDs that are performed when estimating the separate \mathbf{F}_i^w matrices. If one or more of the data blocks \mathbf{X}_i^{within} contain more observations than variables – as is typically the case – computational costs can be decreased by using a speedup that was proposed by Kiers & Harshman (1997) in the context of three-way component analysis. More specifically, one computes the QR decomposition of the \mathbf{X}_i^{within} matrices for which $J < K_i$: $\mathbf{X}_i^{within} = \mathbf{Q}_i \mathbf{R}_i$, with \mathbf{Q}_i a $K_i \times J$ columnwise orthonormal basis matrix and \mathbf{R}_i ($J \times J$) an upper triangular matrix, and replaces \mathbf{X}_i^{within} by \mathbf{R}_i . Analogous to what is explained by Kiers and Harshman (1997), the loss function does not change when replacing \mathbf{X}_i^{within} by \mathbf{R}_i , so minimizing the loss function involving \mathbf{R}_i solves the same problem, but on (much) smaller matrices. Conducting the MLSCA analysis on these smaller matrices yields within-component score matrices $\mathbf{F}_i^{w,red}$ that are reduced in size as well. Afterwards, the within-component score matrices \mathbf{F}_i^w for the full data set are computed based on the $\mathbf{F}_i^{w,red}$ and \mathbf{Q}_i matrices: $\mathbf{F}_i^w = \mathbf{Q}_i \mathbf{F}_i^{w,red}$.

How much computation time can be gained depends of course on the difference between the number of observations per block and the number of variables. To illustrate this, Table 2 first shows the computation time of the within-part of the MLSCA analyses of the cheese data, run on a the same laptop using five within-components, one rational start, and 100 random starts. As the number of observations per block is only slightly larger than the number of variables, there is hardly any gain in computation time. For comparison, Table 2 also displays the computation times for the MLSCA analyses of the cross-cultural data reported by Kuppens et al. (2006), using five within-components and a rational start only. This data set consists of ratings of 14 emotions by inhabitants (observations) of 48 different countries (data blocks). Since the number of inhabitants per block varies between 27 and 549 ($M = 193.75$; $SD = 117.63$), it is no surprise that the analyses are executed much faster when using the computational shortcut.

Model selection

The number of between- (i.e., Q^b) and within- (i.e., Q^w) components and the model variant that is needed to adequately and

Table 2 Computation times (in seconds) of the within-part of MultiLevel Simultaneous Component Analysis (MLSCA) analyses of the cheese data and a cross-cultural data set

Data	Shortcut	Within analysis			
		ECP	IND	PF2	P
Cheese	With	2.732	150.213	161.199	.006
	Without	2.711	157.842	164.737	.005
Cross-cultural	With	.241	9.461	10.668	.030
	Without	1.646	46.357	48.600	.044

parsimoniously summarize the information in a two-level multivariate data set are usually unknown, although prior studies or theory can sometimes yield useful clues. To resolve this model selection problem, one may fit the different MLSCA variants with increasing Q^b and Q^w and use a formal model selection heuristic to assessing the optimal complexity. As the between- and within-part of the data are analyzed independently (see Model subsection above), the appropriate between- and within-submodels can be determined separately.

Between-part

To select the optimal number of between-components, we recommend using the CHull test (Ceulemans & Kiers, 2006; Wilderjans, Ceulemans, & Meers, 2013), which is an extension of the well known scree-test (Cattell, 1966) and has shown good behavior for MLSCA (Ceulemans, Timmerman & Kiers, 2011) as well as for a variety of other model selection problems (Bulteel, Wilderjans, Tuerlinckx, & Ceulemans, 2013; Ceulemans & Kiers, 2009; Ceulemans & Van Mechelen, 2005; Schepers, Ceulemans, & Van Mechelen, 2008). To conduct this test, $VAF\%_{between}$ is first plotted against a complexity measure c_{Q^b} (i.e., the number of free parameters corrected for the number of observations), which can be computed as $[\min(I, \ln(I)J)]Q^b + JQ^b - (Q^b)^2 - Q^b$ (for a detailed explanation, see Ceulemans, Timmerman, & Kiers, 2011). Next, the convex hull of this plot is obtained and the solutions that are located on the higher boundary of this convex hull – denoted as the hull solutions – are retained, as they have the best fit versus complexity balance. Finally, the resulting H hull solutions are scanned to detect the model complexity at which the increase in fit maximally levels off. To determine this point in a more objective way, scree ratios

$$sr_h = \frac{\frac{VAF\%_{0_h}^{between} - VAF\%_{0_{h-1}}^{between}}{c_h - c_{h-1}}}{\frac{VAF\%_{0_{h+1}}^{between} - VAF\%_{0_h}^{between}}{c_{h+1} - c_h}} \quad (8)$$

are computed, with h indicating the h -th hull solution ($h=1..H$), and the solution for which the resulting ratio is

highest is retained (see Ceulemans & Kiers, 2006; Wilderjans, Ceulemans, & Meers, 2013).

When analyzing the cheese data, we fitted between-models with one to six between-components. Applying the CHull procedure reveals that all between-models except for the one with five between-components lie on the higher boundary of the convex hull (see Fig. 1). Based on the scree ratios (see Table 3), a between-model with four between-components is selected.

Within-part

To select the optimal model variant and number of within-components Q^w , one can also apply a CHull test. Specifically, one can compute scree ratios as in equation (8), but using $VAF\%_{within}$ and c_{Q^w} , with the latter value expressing the number of free parameters in the within-submodel, corrected for the number of observations. Table 1 shows the formulas to determine c_{Q^w} (for details, see Ceulemans, Timmerman, & Kiers, 2011). Note, however, that it is not readily clear how the c_{Q^w} values should be adapted to distinguish among the MLSCA-PF2 and MLSCA-PF2-NN variants.

To determine the optimal within-model for the cheese data, analyses with all five model variants (see Model subsection above) were performed with the number of within-components varying from one to six, resulting in 30 within-solutions. Figure 2 shows the resulting hull plot. Applying the numerical convex hull procedure to this plot yielded seven hull solutions. Based on the associated sr_h -values (see Table 4), a MLSCA-PF2 model with two within-components seems indicated.

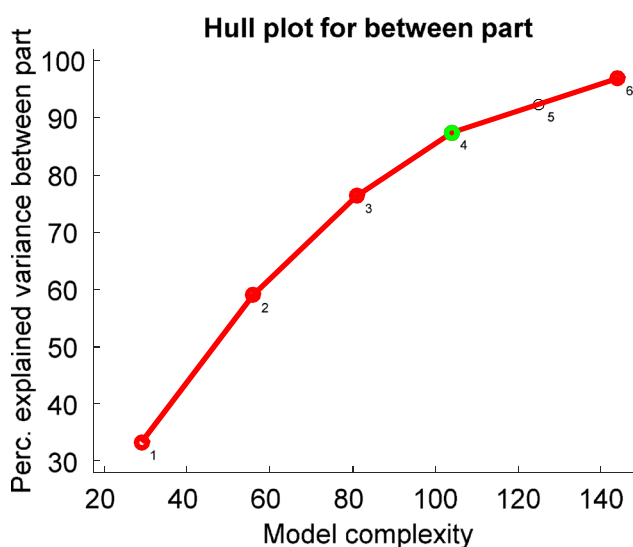


Fig. 1 Hull plot for the between-solutions for the cheese data. The numbers in the plot indicate the complexity of the solutions (i.e., the number of between-components)

Table 3 Fit and complexity values and sr_h -ratio for all between-solutions for the cheese data that are located on the upper boundary of the convex hull

Q^b	c_{Q^b}	$VAF\%^{between}$	sr_h
1	29	33.29	—*
2	56	59.02	1.37
3	81	76.36	1.44
<u>4</u>	<u>104</u>	<u>87.45</u>	<u>2.04</u>
6	144	96.60	—*

*For the first and the last models considered, the sr_h -ratio is not defined
The optimal model is underlined and indicated in italics

Interpreting the parameter estimates of the retained model

Interpretation of the between-submodel

Table 5 displays the raw (i.e., the covariances between the between-components and the between-parts of the observed variables, see above) VARIMAX rotated between-component loadings. Because the between-variance is considerably smaller than the within-variance, the normalized loadings (i.e., correlations rather than covariances; not shown) are much larger, but they are proportionally quite similar to their raw counterparts. From Table 5 it can be concluded that the differences in the mean descriptor profiles of the eight panelists can be summarized by means of the following four dimensions: The first characteristic pertains to chalky (i.e., astringency) and grainy

Table 4 Fit and complexity values and sr_h -ratio for all within-solutions for the cheese data that are located on the upper boundary of the convex hull

Number in plot	Variant	Q^w	c_{Q^w}	$VAF\%^{within}$	sr_h
1	MLSCA-ECP	1	247	24.23	—*
<u>7</u>	<u>MLSCA-PF2</u>	<u>2</u>	<u>501</u>	<u>40.70</u>	<u>1.82</u>
13	MLCA	2	800	51.38	1.18
19	MLCA	3	1176	62.81	1.32
27	MLCA	4	1536	71.10	1.30
28	MLCA	5	1880	77.19	1.26
29	MLCA	6	2208	81.81	—*

*For the first and the last models considered, the sr_h -ratio is not defined
The optimal model is underlined and indicated in italics

taste of the cheese samples, whereas the second characteristic groups fat-related flavor properties (i.e., butter, fat, and creaminess). The other two characteristics pertain to the visual appearance and the sweetness of the cheese samples, respectively. The between-component scores, which reveal the positions that the eight panelists take on these four dimensions, can be read from Table 6. It can, for instance, be derived that the sixth panelist has higher sweetness ratings than the other panelists have.

Interpretation of the within-submodel

Given that the number of observations, and, thus, the number of within-component scores, will often be quite large, the interpretation of the within-model focuses mostly on the within-loadings, which determine the labeling of the within-components.

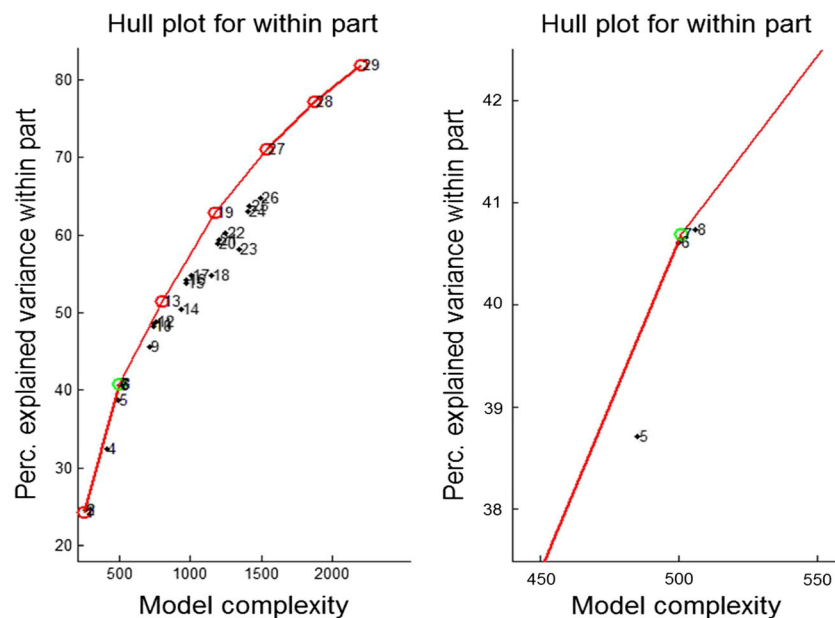


Fig. 2 Hull plot for the within-solutions for the cheese data (left panel); the right panel zooms in on the part of the plot that contains the selected solution. The solutions are ordered from least to most complex

Table 5 Raw between- (VARIMAX rotated) and within-loadings of the MultiLevel Simultaneous Component Analysis (MLSCA)-PF2 model with four between-components and two within-components for the cheese data. Loadings for which the normalized value is larger than or equal to .40 are underlined

Attributes	Between				Within	
	1	2	3	4	1	2
N-old milk	<u>.81</u>	-.08	-.05	-.05	.01	.15
E-green	<u>.71</u>	.12	<u>.41</u>	.09	-.01	-.07
E-grainy	<u>.57</u>	.11	.07	.22	-.06	.09
M-grainy	<u>.46</u>	-.07	<u>.26</u>	.24	.08	-.12
M-chalky	<u>.44</u>	-.16	-.15	-.12	-.10	<u>.58</u>
M-butter	-.05	<u>.60</u>	-.03	.04	-.03	<u>-.63</u>
M-fat	.05	<u>.58</u>	-.13	-.14	-.17	<u>-.57</u>
M-cream	-.08	<u>.56</u>	.03	.07	-.08	<u>-.54</u>
N-butter	.16	<u>.54</u>	-.22	.04	-.03	-.29
M-salt	-.21	<u>.48</u>	-.12	.19	.24	<u>-.41</u>
N-cream	.02	<u>.46</u>	<u>.24</u>	.08	-.10	<u>-.35</u>
E-grey	<u>.24</u>	<u>-.31</u>	<u>.40</u>	-.11	-.11	.18
M-sour	.02	-.19	<u>-.55</u>	-.06	.15	<u>.42</u>
E-white	-.06	.06	<u>-.48</u>	-.03	-.18	<u>.50</u>
M-melt down	.09	.11	<u>-.47</u>	.07	<u>.71</u>	-.23
E-yellow	.18	-.15	<u>.42</u>	.13	.11	<u>-.59</u>
M-sweet	.11	-.03	.01	<u>.75</u>	-.06	-.23
E-shiny	.08	.06	-.11	<u>.30</u>	<u>.65</u>	-.01
M-resistance	-.06	<u>.07</u>	-.03	-.05	<u>-.83</u>	.29
M-creaminess	<u>-.24</u>	.17	.14	.10	-.27	-.32
M-firm	.03	-.03	-.08	<u>-.21</u>	<u>-.84</u>	.24
N-acidic	.06	<u>.29</u>	<u>-.26</u>	<u>-.25</u>	-.02	.12
H-resistance	<u>-.24</u>	-.07	-.03	<u>-.18</u>	<u>-.78</u>	.04

Moreover, depending on the variant, inspecting the block-specific variances and correlations of the within-component scores can also be very informative.

For the cheese example, the within-components capture the main features that each panelist (implicitly or explicitly) uses when judging the separate cheese samples. These components can be labeled by inspecting the raw loadings in Table 5. The MLSCA-PF2 model has no rotational freedom, which implies that the loadings should be interpreted as given in Table 5. The first within-component pertains to texture properties of the cheese (i.e., going from cheese with a firm texture that does not easily melt down in the mouth to soft cheese that breaks down faster), and the second to fat- and color-related properties (i.e., going from fat and creamy yellow cheese to chalky white cheese).

The panelist-specific variances of the within-components, which are shown in Table 6, reveal how salient these components are for each panelist. For instance, it can be concluded that panelists 4, 6, and 7 are relatively variable in their judgments of the texture of the cheeses. The correlation between the within-

Table 6 Between-component scores and standard deviations of the within-component scores of the MultiLevel Simultaneous Component Analysis (MLSCA)-PF2 model with four between-components and two within-components for the cheese data

Panelist	Between-component scores				SDs of within-components	
	1	2	3	4	1	2
Panelist 1	-1.14	-.79	-.63	.55	.84	1.22
Panelist 2	.65	.02	1.53	1.04	.67	1.29
Panelist 3	-.45	1.53	-.15	-1.41	.93	.77
Panelist 4	1.33	-1.53	-1.31	-.58	1.33	1.24
Panelist 5	1.40	1.04	-.10	-.45	.95	.59
Panelist 6	.00	.77	-.66	1.88	1.10	.71
Panelist 7	-1.50	-.05	-.39	-.36	1.10	1.18
Panelist 8	-.29	-1.00	1.71	-.67	.95	.70

components for each single panelist amounts to .11 for three panelists (1, 4, 7) and -.11 for the other five, which implies that the two product features that the judges use are only slightly dependent, with “firm texture” being related to a “yellow color” for panelists 1, 4 and 7 and to “a white color” for the others. The fact that this correlation is weak, implies that the MLSCA-IND model fits the data almost equally well and yields very similar parameter values.

Note that in this example the between-components and within-components pertain to different features. This suggests that features relevant for describing structural differences between raters (as captured in the between-loadings) differ from those relevant for describing structural differences between the cheeses (as captured in the within-loadings). In other applications, the between- and within-features may or may not differ substantially.

The MLSCA software package

The MLSCA software can be downloaded from <http://ppw.kuleuven.be/okp/software/> MLSCA. The package includes all the files that are necessary to replicate the analysis of the cheese data, so that users can get a sense of how the program works, and of what the output looks like and can be interpreted.

Users interact with the program through a graphic user interface (GUI), which is built around a set of MATLAB m-files. Two versions of the software are available: a standalone application, which runs on any Windows computer and does not require MATLAB to be installed, and a MATLAB application. Whereas the standalone version can be launched from the Windows start menu (icon *MLSCA_gui.exe*; see also installation manual at the website mentioned above), the MATLAB version can be started by setting the current MATLAB directory to the folder in which all the software files are stored and typing *MLSCA_gui* at the command prompt. The GUI that appears (see Fig. 3) consists of

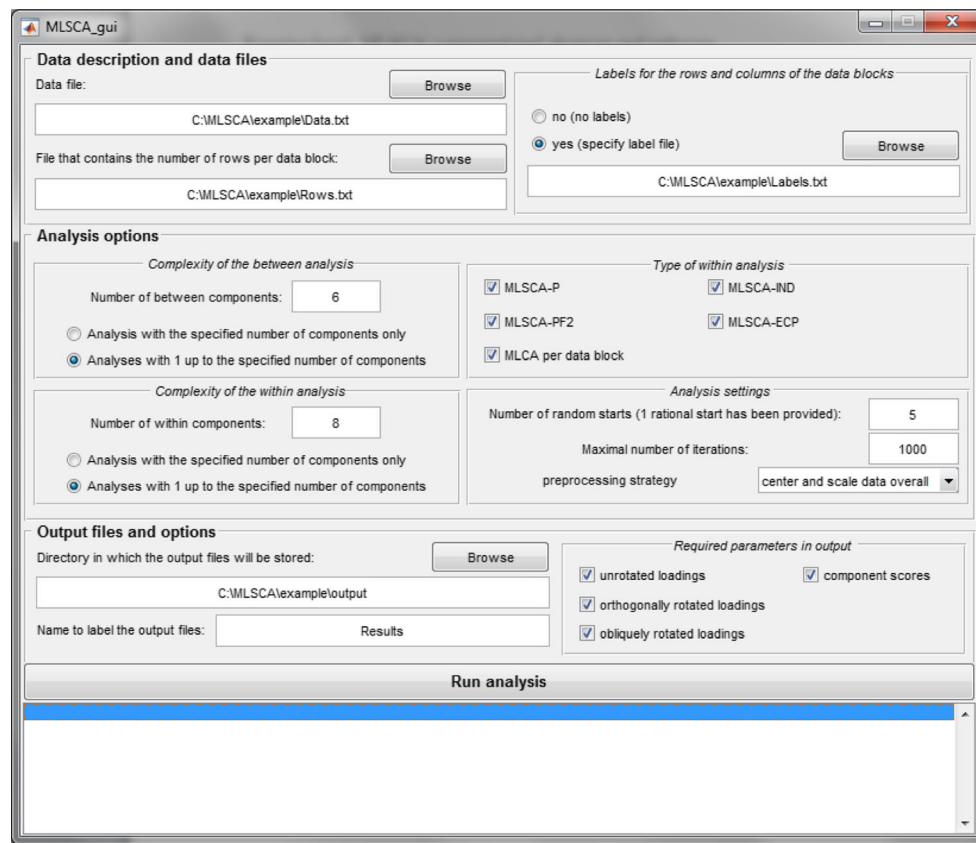


Fig. 3 Screenshot of the MultiLevel Simultaneous Component Analysis (MLSCA) graphic user interface

three compartments: Data description and data files, Analysis options, and Output files and options. To perform a ML(S)CA analysis, the user specifies the necessary information in the different compartments and clicks the *Run analysis*-button. In the following paragraphs, the three compartments will be outlined, closing off with error handling.

Data description and data files

Data file

First, the user specifies the data file by means of the “Data file” *Browse*-button. For instance, from Fig. 3, it can be read that the cheese data are stored in *Data.txt*; an excerpt of this file is displayed in the left-hand panel of Fig. 4. The data file should be an ASCII file (i.e., *.txt* file) that contains a row for each observation and is sorted according to the blocks. No empty lines are allowed between the rows of a single data block. However, rows belonging to different data blocks may be separated by one or more empty lines. Per row, the scores on the variables are separated by one or more spaces, commas, semicolons, or tabs, or any combination of these. Missing data are not allowed. Each observed score should be an integer or a real number, with decimal separators being denoted by a period and *not* by a comma!

Data description

To provide information regarding the number of observations in each data block, the user clicks the *File that contains the number of rows per data block* *Browse*-button and selects the appropriate file (for our example: *Rows.txt*, see right-hand panel of Fig. 4). This file should again be of the ASCII format, containing as many rows as there are data blocks; each row consists of a single number indicating the number of observations within the corresponding data block.

Label file

Optionally, the user may provide labels for the observations, variables, and data blocks. To this end, the user checks the option *yes (specify the labels)* and, subsequently, browses for the ASCII file containing the labels (for our example: *Labels.txt*, see Fig. 3; the content of this file is shown in the middle panel of Fig. 4). The label file contains three sets of labels, for the data blocks, block-observation combinations, and variables, respectively. Each label should be placed on a separate line. The different sets may be separated by one or more empty lines; however, within each set, only line breaks are allowed. Obviously, the number and ordering of the labels

Type of within analysis

The user may select one or more within-submodel variants (see Model subsection above): (1) MLCA, (2) MLSCA-P, (3) MLSCA-IND, (4) MLSCA-PF2, and (5) MLSCA-ECP.

Analysis settings

When fitting MLSCA-IND, MLSCA-PF2, and/or MLSCA-ECP, an ALS algorithm is adopted. In order to minimize the risk of obtaining a local minimum only, a multi-start procedure is used consisting of one rational and a number of random starts (see Algorithm subsection above). The user may alter the number of random starts (five by default) that needs to be used by entering a number (integer) in the box next to the *Number of random starts (1 rational start has been provided)* field. Further, the user may indicate the maximal number of ALS iterations (1,000 by default) that will be performed by means of the *Maximal number of iterations* box. Note, however, that if this maximum number of iterations is reached, this indicates that the obtained solution should be interpreted with much caution as the algorithm did not converge properly. Finally, the user has to decide about the *preprocessing strategy* by selecting whether (*center and scale data overall* option) or not (*center data overall* option) the variables should be scaled to a variance of one across blocks. Hence, the software always grand-mean centers the data.

Output files and options

In the *Output files and options* compartment (see Fig. 3), the user selects the directory (for the cheese example “C:\MLSCA\example\output”) where all the output files have to be stored, by means of the corresponding *Browse* button. Furthermore, the user specifies a string (*Name to label the output files*) which will constitute the first part of the name of all output files (i.e., for the cheese example “Results”). Finally, the user checks the appropriate boxes in the *Required parameters in output* panel, indicating which parameter estimates need to be presented in the output: (a) unrotated loadings, (b) orthogonally rotated loadings, using the VARIMAX criterion, (c) obliquely rotated loadings, on the basis of the HKIC criterion (Kiers & ten Berge, 1994b), and/or (d) the associated component scores.

When the analysis is finished, multiple output files with different extensions (i.e., *.mht*, *.txt*, and *.mat*) are stored in the selected output folder. The information in the *.txt* files enables the user to load the results easily into any popular software package, like SAS and SPSS, in order to further

process the results. In the remainder, the content of the different output files will be described.

Output file with *.mht* extension

The *.mht* file (in our example “Results_overview.mht”) contains a summary of the analysis. Specifically, first, information regarding the amount of between-block and within-block variation in the data is displayed. Second, the percentage of between- and within-variation that is explained by the fitted between- and within-submodels is shown, along with a between- and within-Variance Accounted For (VAF) plot and model selection advice. Finally, for each considered within-variant, the total percentage of within-variance explained by the different solutions is displayed for each data block separately.

Output files with *.txt* extension

When the user checked loadings and/or component scores in the *Required parameters in output* panel, the between- and within-solutions are stored in separate files, of which the name indicates the model variant (i.e., between, PCA, SCAP, SCAI ND, SCAPF2, or SCAECP) and the rotation used (i.e., *_unrotated*, *_rotated*, and *_oblique*). If unrestricted, the component variances and correlations per data block are also displayed.

Output files with *.mat* extension (for MATLAB version only)

These files contain an object containing the results of the different analyses.

Status of the analysis and error handling

Once the user has specified the necessary input and output files and analysis options, the analysis can be started by clicking the *Run analysis*-button. During the analysis, the MLSCA program displays information concerning the status of the analysis in the box at the bottom of the GUI screen (see Fig. 3). When the analysis has finished, a screen will appear notifying the user that *The analysis has been finished!*. When the input files do not comply with the requirements mentioned above, the analysis will be deferred and the MLSCA program will produce one or more error screens, which will contain information regarding the encountered problem(s). To aid the user in dealing with the encountered problem(s), the content of the error message(s) will also be displayed in the box at the bottom of the GUI screen. Once all specifications are corrected, the user may click the *Run analysis*-button to restart the analysis.

Concluding remarks

In this paper, we discussed a computational shortcut for MLSCA. Further, we presented a software package, MLSCA, to perform MLSCA analyses. Three comments need to be made.

First, as MLSCA is a deterministic approach, no statistical tests or estimates of the standard errors of the parameters are provided. Yet, to get insight into the reliability of the estimates, one may adopt a bootstrap procedure to obtain confidence intervals for the ML(S)CA parameters (see Timmerman, Kiers, Smilde, Ceulemans, & Stouten, 2009). As many options have to be specified (e.g., which data levels are considered random), leading to rather different bootstrap procedures and analysis output, we opted to not include it in the MLSCA program. However, the MATLAB code for performing such a bootstrap analysis is available upon request. Moreover, we will explore how this option can be incorporated efficiently and in a user-friendly manner in a future version of the software.

Second, as for regular component analysis (Hubert, Rousseeuw, & Vanden Branden, 2005), MLSCA results can be affected by bad leverage outliers. To trace such outliers and obtain robust estimates, Ceulemans, Hubert and Rousseeuw (2013) presented a robust version of MLSCA-P. As the associated estimation procedure relies heavily on the Libra toolbox and is only available for MLSCA-P, the robust approach cannot be incorporated in the software package, but m-files can be provided upon request.

Finally, two-level multivariate data often contain missing data. Up to now, this problem is in most cases dealt with by list-wise deletion of the corresponding observations (see e.g., Kuppens et al., 2006), which may yield biased results. Recently, Josse, Timmerman and Kiers (2013) proposed an imputation approach for the least restrictive MLSCA variant. However, for the moment it is unclear how missing data have to be treated when using the other MLSCA variants. Therefore, we did not include this imputation approach in the current version of the package.

Acknowledgments The research reported in this paper was partially supported by the Fund for Scientific Research (FWO)-Flanders (G.0582.14), the Research Fund of KU Leuven (GOA/15/003), and the Interuniversity Attraction Poles programme financed by the Belgian government (IAP/P7/06). Tom Wilderjans is a post-doctoral fellow of the Fund for Scientific Research (FWO)-Flanders.

References

- Bosmans, G., Van de Walle, M., Goossens, L., & Ceulemans, E. (2014). (In)variability of attachment in middle childhood: Secure base script evidence in diary data. *Behavior Change*, 31(4), 225–242.
- Bro, R., Qannari, E. M., Kiers, H. A. L., Naes, T., & Frøst, M. B. (2008). Multi-way models for sensory profiling data. *Journal of Chemometrics*, 22, 36–45.
- Brose, A., De Roover, K., Ceulemans, E., & Kuppens, P. (2015). Older adults' affective experiences across 100 days are less variable and less complex than younger adults'. *Psychology and aging*, 30, 194–208.
- Bulteel, K., Wilderjans, T. F., Tuerlinckx, F., & Ceulemans, E. (2013). CHull as an alternative to AIC and BIC in the context of mixtures of factor analyzers. *Behavior Research Methods*, 45, 782–791.
- Cattell, R. B. (1966). The meaning and strategic use of factor analysis. In R. B. Cattell (Ed.), *Handbook of multivariate experimental psychology* (pp. 174–243). Chicago, US: Rand McNally.
- Ceulemans, E., Hubert, M., & Rousseeuw, P. (2013). Robust multilevel simultaneous component analysis. *Chemometrics and Intelligent Laboratory Systems*, 129, 33–39.
- Ceulemans, E., & Kiers, H. A. L. (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology*, 59, 133–150.
- Ceulemans, E., & Kiers, H. A. L. (2009). Discriminating between strong and weak structures in three-mode principal component analysis. *British Journal of Mathematical and Statistical Psychology*, 62, 601–620.
- Ceulemans, E., Timmerman, M. E., & Kiers, H. A. L. (2011). The CHull procedure for selecting among multilevel component solutions. *Chemometrics and Intelligent Laboratory Systems*, 106, 12–20.
- Ceulemans, E., & Van Mechelen, I. (2005). Hierarchical classes models for three-way three-mode binary data: Interrelations and model selection. *Psychometrika*, 70, 461–480.
- De Roover, K., Ceulemans, E., & Timmerman, M. E. (2012). How to perform multiblock component analysis in practice. *Behavior Research Methods*, 44, 41–56.
- De Roover, K., Timmerman, M. E., Mesquita, B., & Ceulemans, E. (2013a). Common and cluster-specific simultaneous component analysis. *PLoS ONE*, 8(e62280), 1–14.
- De Roover, K., Timmerman, M. E., Van Diest, I., Onghena, P., & Ceulemans, E. (2014). Switching principal component analysis for modeling changes in means and covariance structure over time. *Psychological Methods*, 19, 113–132.
- De Roover, K., Timmerman, M. E., Van Mechelen, I., & Ceulemans, E. (2013b). On the added value of multiset methods for three-way data analysis. *Chemometrics and Intelligent Laboratory Systems*, 129, 98–107.
- Erbas, Y., Ceulemans, E., Boonen, J., Noens, I., & Kuppens, P. (2013). Emotion differentiation in autism spectrum disorder. *Research in Autism Spectrum Disorders*, 7, 1221–1227.
- Erbas, Y., Ceulemans, E., Pe, M., Koval, P., & Kuppens, P. (2014). Negative emotion differentiation: Its personality and well-being correlates and a comparison of different assessment methods. *Cognition and Emotion*, 28, 1196–1213.
- Goldstein, H., & Browne, W. J. (2005). Multilevel factor analysis models for continuous and discrete data. In A. Maydeu-Olivares & J. J. McArdle (Eds.), *Contemporary psychometrics: a festschrift for Roderick P. McDonald* (pp. 453–475). New Jersey, US: Lawrence Erlbaum.
- Harshman, R. A. (1972). PARAFAC2: Mathematical and technical notes. *UCLA Working Papers in Phonetics*, 22, 30–44.
- Harshman, R. A., & De Sarbo, W. S. (1984). An application of PARAFAC to a small sample problem, demonstrating preprocessing, orthogonality constraints, and split-half diagnostic techniques. In H. G. Law, C. W. Snyder Jr., J. A. Hattie, & R. P. McDonald (Eds.), *Research methods for multimode data analysis* (pp. 602–642). New York: Praeger.
- Helwig, N. E. (2013). The special sign indeterminacy of the direct-fitting Parafac2 model: Some implications, cautions, and recommendations

- for Simultaneous Component Analysis. *Psychometrika*, 78, 725–739.
- Hubert, M., Rousseeuw, P., & Vanden Branden, K. (2005). ROBPCA: A new approach to robust principal components analysis. *Technometrics*, 47, 64–79.
- Jansen, J. J., Hoefsloot, H. C. J., van der Greef, J., Timmerman, M. E., & Smilde, A. K. (2005). Multilevel component analysis of time-resolved metabolic fingerprinting. *Analytica Chimica Acta*, 530, 173–183.
- Josse, J., Timmerman, M. E., & Kiers, H. A. L. (2013). Missing values in multi-level simultaneous component analysis. *Chemometrics and Intelligent Laboratory Systems*, 129, 21–32.
- Kiers, H. A. L., & Harshman, R. A. (1997). Relating two proposed methods for speedup of algorithms for fitting two- and three-way principal component and related multilinear models. *Chemometrics and Intelligent Laboratory Systems*, 36, 31–40.
- Kiers, H. A. L., & ten Berge, J. M. F. (1994a). Hierarchical relations between methods for Simultaneous Component Analysis and a technique for rotation to a simple simultaneous structure. *British Journal of Mathematical and Statistical Psychology*, 47, 109–126.
- Kiers, H. A. L., & ten Berge, J. M. F. (1994b). The Harris-Kaiser independent cluster rotation as a method for rotation to simple component weights. *Psychometrika*, 59, 81–90.
- Kiers, H. A. L., ten Berge, J. M. F., & Bro, R. (1999). PARAFAC2—part I: A direct-fitting algorithm for the PARAFAC2 model. *Journal of Chemometrics*, 13, 275–294.
- Kuppens, P., Ceulemans, E., Timmerman, M. E., Diener, E., & Kim-Prieto, C. Y. (2006). Universal intracultural, and intercultural dimensions of the recalled frequency of emotional experience. *Journal of Cross Cultural Psychology*, 37, 491–515.
- Mehta, P. D., & Neale, M. C. (2005). People are variables too: Multilevel structural equations modeling. *Psychological Methods*, 10, 259–284.
- Schepers, J., Ceulemans, E., & Van Mechelen, I. (2008). Selecting among multi-mode partitioning models of different complexities: A comparison of four model selection criteria. *Journal of Classification*, 25, 67–85.
- Snijders, T. A. B., & Bosker, R. (1999). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. London: Sage Publishers.
- Stouten, J., Ceulemans, E., Timmerman, M. E., & Van Hiel, A. (2011). Tolerance of justice violations: The effects of need on emotional reactions after violating equality in social dilemmas. *Journal of Applied Social Psychology*, 41, 357–380.
- Timmerman, M. E. (2006). Multilevel component analysis. *British Journal of Mathematical and Statistical Psychology*, 59, 301–320.
- Timmerman, M. E., Ceulemans, E., Lichtwarck-Aschoff, A., & Vansteelandt, K. (2009a). Multilevel component analysis for studying intra-individual variability and inter-individual differences. In J. Valsiner, P. C. M. Molenaar, M. C. D. P. Lyra, & N. Chaudhary (Eds.), *Dynamic process methodology in the social and developmental sciences* (pp. 291–318). New York, US: Springer.
- Timmerman, M. E., & Kiers, H. A. L. (2003). Four simultaneous component models of multivariate time series for more than one subject to model intraindividual and interindividual differences. *Psychometrika*, 68, 105–122.
- Timmerman, M. E., Kiers, H. A. L., Smilde, A. K., Ceulemans, E., & Stouten, J. (2009b). Bootstrap confidence intervals in Multi-level simultaneous component analysis. *British Journal of Mathematical and Statistical Psychology*, 62, 299–318.
- Wilderjans, T. F., & Ceulemans, E. (2013). Clusterwise Parafac to identify heterogeneity in three-way data. *Chemometrics and Intelligent Laboratory Systems*, 129, 87–97.
- Wilderjans, T. F., Ceulemans, E., & Meers, K. (2013). CHull: A generic convex hull based model selection method. *Behavior Research Methods*, 45, 1–15.